

What is claimed is:

1. System for providing traffic emulation for packet-switched networks, said system comprising:

an endpoint; and

5 an emulator module associated with said endpoint;

said emulator module comprising at least one finite state machine for modeling traffic flows to be emulated.

2. The system according to Claim 1, wherein said emulator module comprises an emulation manager, said emulation manager comprising a finite state machine for
10 maintaining the status of an emulation operation.

3. The system according to Claim 2, wherein said finite state machine of said emulation manager comprises three stages, said three stages comprising an initialization stage, and emulation stage and a result reporting stage.

4. The system according to Claim 1, wherein:
15 said endpoint further comprises a network interface module for facilitating the communication of flows to a network; and

said at least one finite state machine comprises a plurality of finite state machines, wherein each finite state machine corresponds to a different one of said flows.

5. The system according to Claim 4, wherein said network interface module comprises:

at least one port for receiving data;

at least one port for receiving signals;

5 a first background process for managing the receipt of data at said at least one data port; and

a second background process for managing the receipt of signals at said at least one signal port.

6. The system according to Claim 1, wherein said emulator module further
10 comprises an event scheduler, said event scheduler comprising an event queue.

7. Method of providing traffic emulation for packet-switched networks, said method comprising:

providing an endpoint; and

providing an emulator module associated with said endpoint;

15 said step of providing an emulator module comprising providing at least one finite state machine for modeling traffic flows to be emulated; and

modeling traffic flows with said at least one finite state machine.

8. The method according to Claim 7, wherein:

said step of providing an emulator module comprises providing an emulation manager;

said step of providing an emulation manager comprises providing a finite state machine for maintaining the status of said emulation; and

5 said method further comprising maintaining the status of said emulation with said finite state machine of said emulation manager.

9 The method according to Claim 8, wherein:

said step of providing a finite state machine for maintaining the status of said emulation comprises providing three stages, said three stages comprising an initialization stage, and
10 emulation stage and a result reporting stage; and

said method further comprises:

performing each of said three stages in the following sequence: initialization stage, emulation stage and result reporting stage; and

thereafter returning to said initialization stage.

15 10. The method according to Claim 7, wherein:

said step or providing an endpoint comprises providing a network interface module for facilitating the communication of flows to a network; and

said step of providing at least one finite state machine comprises providing a plurality of finite state machines, wherein each said finite state machine corresponds to a different one of said flows.

11. The method according to Claim 10, wherein said step of providing a network
5 interface module comprises providing:

at least one port for receiving data;

at least one port for receiving signals;

a first background process for managing the receipt of data at said at least one data
port; and

10 a second background process for managing the receipt of signals at said at least
one signal port;

said method further comprising:

managing the receipt of data at said at least one data port with said first
background process; and

15 managing the receipt of signals at said at least one signal port with said second
background process.

12. The method according to Claim 7, wherein said step of providing an emulator
module comprises providing an event scheduler, said event scheduler comprising an event
queue.